The physical world has a dedicated function to cast rays into the environment - `RayWorldIntersection`. Depending on the material the ray hits and the ray properties, a hit might be "pierceable" or solid. More specifically, a pierceable hit is a hit that has its material pierceability strictly higher than the ray's pierceability (material pierceability and ray pirceability occupy the lowest 4 bits of material flags and RayWorldIntersection flags correspondingly). Pierceable hits don't stop the ray and are accumulated as a sorted (by hit distance) list. The caller provides the function with an array for the hits. A solid hit always takes the slot with index 0 and pierceable hits - slots from 1 to the end. Optionally, the function can separate between 'important' and 'unimportant' pierceable hits (importance is indicated by *sf_important* in material flags) and can make all important hits have higher priority (regardless of hit distance) than unimportant ones when competing for space in the array. By default RayWorldIntersection checks only entity parts with the *geom_colltype_ray* flag, but it is possible to specify another flag or combination of flags (in this case all flags should be set in part for it to be tested) by setting rwi_pierceability() in flags. `RayTraceEntity` is a more low-level function and checks ray hits with a single entity. It returns only the closest hit.

In addition to rays, the physics support geometry casts with `PrimitiveWorldIntersection` function. A geometry cast checks how far a geometry can be moved along a specific line before it collides with any physicalized geometry. All primitives types and a single triangle are supported as input. Note that backward tracing is not supported, so if cast geometry intersects with something initially, that contact will not be detected and the geometry will be traced further until it find another contacting features. Optionally, `PrimitiveWorldIntersection` can do static (i.e. not swept along a line) intersection tests. Note that intersection check only checks for surface intersections, not full containment. Containment can be checked separately, either with pe_status_contains_point or via IGeometry interface.

An important difference between ray and primitive casts is that the former one checks entity grid cells that are touched by the ray, and the latter checks the axis-aligned bounding box of the entire cast, so it shouldn't be used for long casts.

Both casts can be either immediate or queued. Immediate casts are executed right when the function is called, in the calling thread. Queued casts are accumulated and executed in the physics thread before the next physical world step. Results are reported via physical events (EventPhysRWIResult and EventPhysPWIResult).